1.3 Basic Arithmetic and coding in R

- **Assign value to an object**

  (1) use the equal sign =

```
> x = 16
> print(x)
[1] 16
```

  (2) use an arrow <-

```
> x <- 10
> print(x)
[1] 10
```

* Object name can include numbers or periods, but cannot begin with numbers

* To assign characters to object, use " ". Numbers in quotation marks will be treated as characters and not numeric when performing operation.

- **Print value of an object**

  (1) use print( )

  (2) type the object name itself

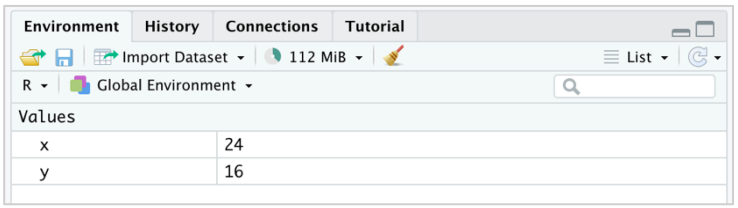
```
> x = 24
> print(x)
[1] 24
> x
[1] 24
```

* R is case sensitive, object y is not equal to Y

```
> y = 16
> print(Y)
錯誤發生在 print(Y)：找不到物件 'Y'
```

- **To know what is stored in R memory**

  (1) look up "workspace" in R studio



  (2) use ls command

```
> ls()
[1] "x" "y"
```

- **Remove an object from workspace memory**

  (1) use rm command

```
> rm(x)
> print(x)
錯誤發生在 print(x)：找不到物件 'x'
```

- **Basic arithmetic**

加	x + y	乘	x * y	次方	x^2	對數	log(x)
減	x - y	除	x / y	根號	sqrt(x)	絕對值	abs(x)

- **其他技巧**

  (1) If an incomplete command has been entered, R will follow that up with a plus sign to remind you. Use “esc” to escape.

```
> sqrt(y
+
+
+ )
[1] 4
```

  (2) Using the “arrow up key” on the keyboard will bring you to the latest command that was entered in R. Hitting it again will bring you to the previously entered command and so on.

  (3) 用 # 加註解

```
> hello World
錯誤: 未預期的符號 in "hello World"
> #hello world
>
```

- **Working directory (預設路徑) :**

  getwd()

- **Create vector (向量) :**

  (1) c(, ,)

```
> newObject <- c(5,12,36)
>
```

  (2)

  * 向量內要求資料型態一致

- **To know the function of a command:**

  (1) ? + command_name

  (2) help(command_name)

  結果會顯示在右下方的 HELP

1.4 Create and Work with Vectors and Matrices

- **Create a vector (column of number or characters)**

  (1) use “c” or concatenate command

```
> x1 <- c(1,3,5,7,9)
> x1
[1] 1 3 5 7 9
```

数字

```
> gender <- c("male", "female")
> gender
[1] "male"   "female"
```

字串

- **Create a sequence**

  (1) create only integer values, use a:b

```
> 2:10
[1]  2  3  4  5  6  7  8  9 10
```

  (2) for more general sequences, use “seq” command: seq(from=a, to=b, by=c)

```
> seq(from=1, to=5, by=1/4)
 [1] 1.00 1.25 1.50 1.75 2.00 2.25 2.50 2.75 3.00 3.25 3.50 3.75 4.00 4.25 4.50 4.75
[17] 5.00
>
```

by 代表每次增加多少

- **Create a vector of repeated numbers or characters**

  (1) use the “rep” command: rep(a, times=b)

```
> rep(18, times=5)
[1] 18 18 18 18 18
```

```
> rep("good", times=3)
[1] "good" "good" "good"
```

  (2) have a sequence of numbers repeated multiple times

```
> rep(1:3, times=4)
 [1] 1 2 3 1 2 3 1 2 3 1 2 3
```

整數

```
> rep(seq(from=1, to=3, by=0.5), times=3)
 [1] 1.0 1.5 2.0 2.5 3.0 1.0 1.5 2.0 2.5 3.0 1.0 1.5 2.0 2.5 3.0
```

非整數

  (3) have a sequence of characters repeated multiple times

```
> rep(c("true","false"), times=2)
[1] "true"  "false" "true"  "false"
```

- **Basic calculation**

  x = 1:5, y = c(1,3,5,7,9)

  (1) 加減乘除 to each element of the vector

```
> x + 10
[1] 11 12 13 14 15
```

```
> x - 10
[1] -9 -8 -7 -6 -5
```

```
> x*10
[1] 10 20 30 40 50
```

```
> x/2
[1] 0.5 1.0 1.5 2.0 2.5
```

(2) if two vectors are of the same length, we can add/subtract/multiply/divide corresponding elements

```
> x
[1] 1 2 3 4 5
> y
[1] 1 3 5 7 9
```

```
> x+y
[1]  2  5  8 11 14
```

```
> x-y
[1]  0 -1 -2 -3 -4
```

```
> x*y
[1]  1  6 15 28 45
```

```
> x/y
[1] 1.0000000 0.6666667 0.6000000 0.5714286 0.5555556
```

- **Extract elements of a vector**

  (1) use object_name[a]

```
> y[3]
[1] 5
```

  (2) to extract all elements except a, use object_name[-a]

```
> y[-3]
[1] 1 3 7 9
```

  (3) to extract only some elements (combine: concept of sequence and vector)

```
> y[2:4]
[1] 3 5 7
```

```
> y[c(1,5)]
[1] 1 9
```

```
> y[-c(1,5)]
[1] 3 5 7
```

  (4) to extract only the elements in certain range

```
> y[y<6]
[1] 1 3 5
```

- **Create a matrix of values**

  (1) use “matrix” command: matrix(value, nrow=a, nrow=b, byrow=TRUE/FALSE)

```
> matrix(1:9, nrow=3, byrow=TRUE)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

* 如果把 matrix 裡面某個欄位的值指派為 character，那整個 matrix 都會變成 character

```
> matrix(c(1,2,3,4,5,6,7,8,9), nrow=3, byrow=FALSE)
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

byrow 控制排列方向
TRUE: 依列排序
FALSE: 依欄排序

(2) extract elements from matrix: matrix_name[row_number, column_number]

```
> mat1
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

```
> mat1[1,2]
[1] 2
```

第一列
第二欄

```
>  mat1[c(1,3),2]
[1] 2 8
```

```
> mat1[2, ]
[1] 4 5 6
```

```
> mat1[ ,3]
[1] 3 6 9
```

(3) perform element-wise 加減乘除（跟 vector 一樣）

```
> mat1*10
     [,1] [,2] [,3]
[1,]   10   20   30
[2,]   40   50   60
[3,]   70   80   90
```

1.5 Import Data, Copy Data from Excel to R CSV &TXT Files

*Two options: save the imported data file as a) .csv, comma separated value
b) .txt, tab delimited text file

a) To import .csv file

(1) use the “read.csv” command: read.csv(file.choose(), header=T/F)

* 利用 file.choose 會跳出視窗讓你選要輸入的檔案，不用複製路徑等等
Header 表示原始資料的第一列是否為變數名稱

```
> data1=read.csv(file.choose(), header=T)
> data1
    LungCap Age Height Smoke Gender Caesarean
1     6.475   6   62.1    no   male        no
2    10.125  18   74.7   yes female        no
3     9.550  16   69.7    no female       yes
4    11.125  14   71.0    no   male        no
5     4.800   5   56.9    no   male        no
```

(2) use the “read.table” command: read.table(file.choose(), header=T/F, sep=”,”)

* sep(aration) 代表檔案是用什麼符號分隔，csv 檔就是逗號，txt 檔就是\t

```
> data2=read.table(file.choose(), header=T, sep=",")
> data2
    LungCap Age Height Smoke Gender Caesarean
1     6.475   6   62.1    no   male        no
2    10.125  18   74.7   yes female        no
3     9.550  16   69.7    no female       yes
4    11.125  14   71.0    no   male        no
5     4.800   5   56.9    no   male        no
```

b) To import .txt file

(1) use the “read.delim” command: read.delim(file.choose(), header=T/F)

```
> data3=read.delim(file.choose(), header=TRUE)
> data3
    LungCap Age Height Smoke Gender Caesarean
1     6.475   6   62.1    no   male        no
2    10.125  18   74.7   yes female        no
3     9.550  16   69.7    no female       yes
4    11.125  14   71.0    no   male        no
5     4.800   5   56.9    no   male        no
```

(2) use “read.table” command: read.table(file.choose(), header=T/F, sep=”\t”)

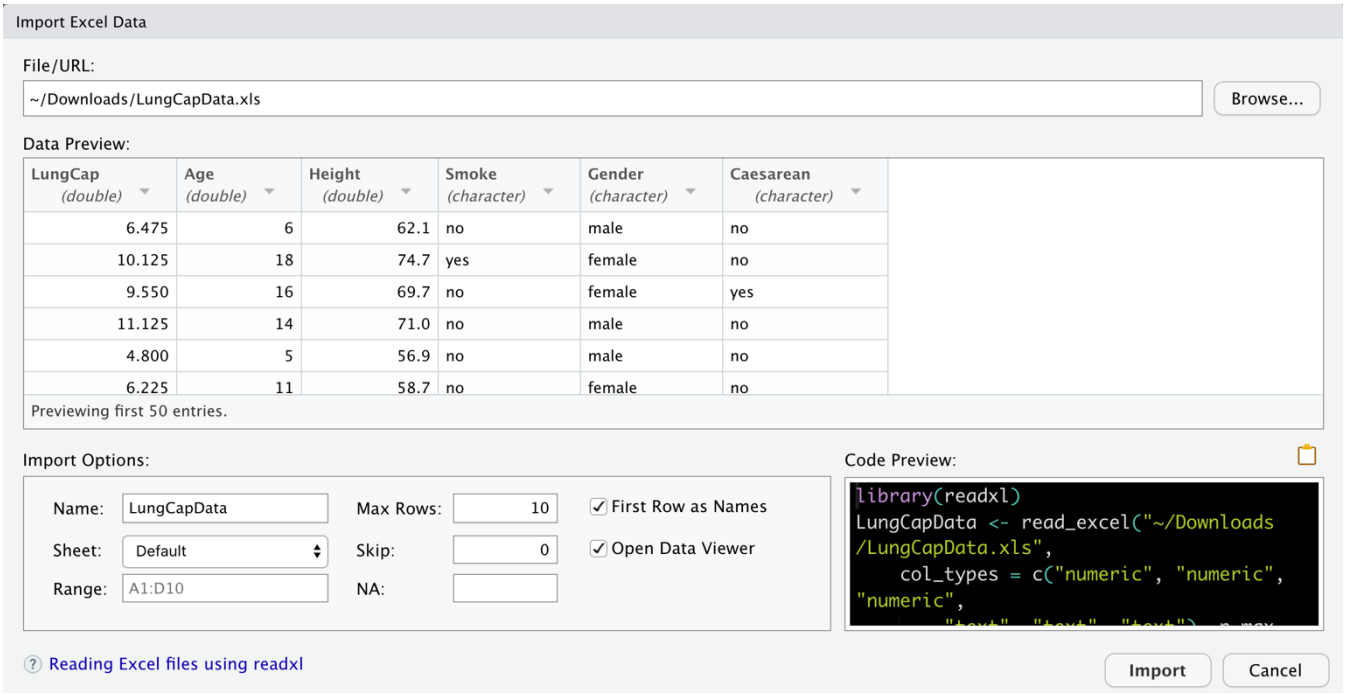
```
> data4 <- read.table(file.choose(), header=T, sep="\t")
> data4
    LungCap Age Height Smoke Gender Caesarean
1     6.475   6   62.1    no   male        no
2    10.125  18   74.7   yes female        no
3     9.550  16   69.7    no female       yes
4    11.125  14   71.0    no   male        no
5     4.800   5   56.9    no   male        no
```

1.6 Importing/Reading Excel Data into R: Using RStudio (readxl)

*The readxl package (pre-install in R) can import both .xlsx and .xls files.

*Two ways: (a) File > Import Dataset > From Excel

(b) Import (右上角 in Environment) > From Excel , demonstrated here



(1) File/URL: 如果想要直接從網路上 import data，貼網址在這欄

(2) Browse: 從電腦上的檔案中 import data

(3) Name: data import 進來之後想存的名字，預設為原檔名

(4) Sheet: 要 import excel 檔案中的哪一個分頁

(5) Range: 原檔的範圍（有一些欄位不想 import 進來的時候可以用）

(6) Max rows: 最多 import 幾列（想限制 data 數量時可以用）

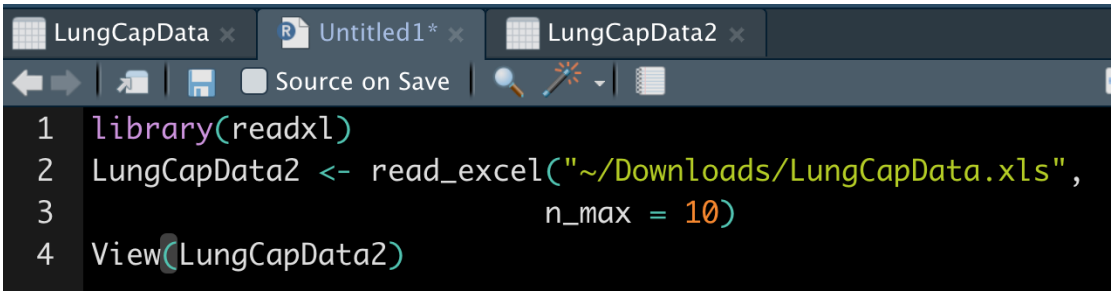
(7) skip: 要跳過前面幾列不 import（填 1 的話跳過第一列，2 的話跳過第一二列）

(8) NA: 原檔中 N/A 或缺值的欄位是以什麼符號表示

(9) Preview 裡面 variable name 下面的小三角形: import 進來時用什麼 class 存那一欄

(10) Code Preview 旁邊的剪貼簿符號: 按一下，未來 import 同個檔案就會用這個預設

(11) Code Preview: 複製起來貼到 RScript 一樣可以用程式碼方式 import



1.7 Export Data from R (csv, txt and other formats)

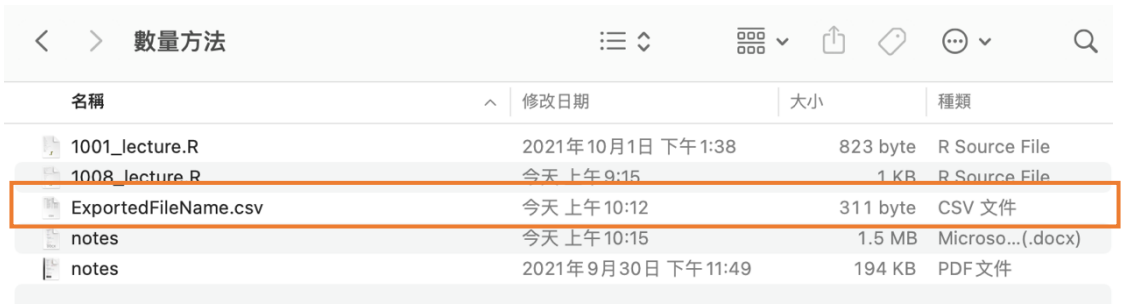
```
> datatoexport
          Subject Age Gender Score
1 Dave.Andreychuk  53   male  80.5
2     Jon.Stewart  54   male  82.1
3        Jane.Doe  38 female  75.9
4  Amelia.Earhart 119 female  90.0
5    Donald.Trump  70   male -25.5
6   Sidney.Crosby  28   male  87.2
7   Oprah.Winfrey  62 female  88.8
8      Steve.Jobs  61   male  91.1
```

(1) use “write.table” command (the most flexible to export data):

write.table(要輸出的 data name, file=”輸出後存成什麼檔名”, sep=” ”)

*會存在 current working directory

```
3  write.table(datatoexport, file="ExportedFileName.csv", sep=",")
```



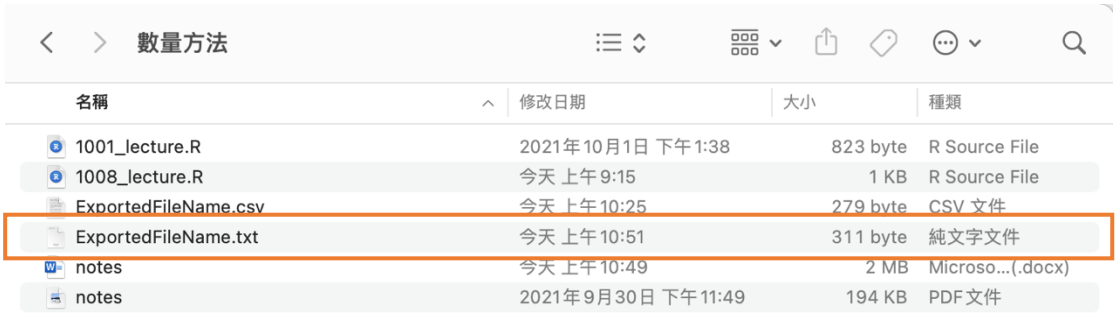
	A	B	C	D	E
1	Subject	Age	Gender	Score	
2	1	Dave.Andrey	53	male	80.5
3	2	Jon.Stewart	54	male	82.1
4	3	Jane.Doe	38	female	75.9
5	4	Amelia.Earha	119	female	90
6	5	Donald.Trum	70	male	-25.5
7	6	Sidney.Crosb	28	male	87.2
8	7	Oprah.Winfr	62	female	88.8
9	8	Steve.Jobs	61	male	91.1

* 打開存好的檔案會發現最左邊新增了一欄列數，加上 row.names=F 就會去掉

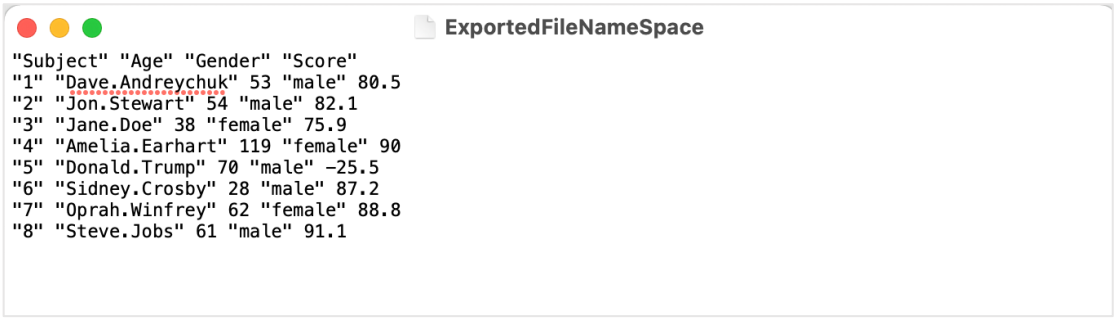
```
write.table(datatoexport, file="ExportedFileName.csv", row.names=F, sep=",")
```

	A	B	C	D
1	Subject	Age	Gender	Score
2	Dave.Andrey	53	male	80.5
3	Jon.Stewart	54	male	82.1
4	Jane.Doe	38	female	75.9
5	Amelia.Earha	119	female	90
6	Donald.Trum	70	male	-25.5
7	Sidney.Crosb	28	male	87.2
8	Oprah.Winfr	62	female	88.8
9	Steve.Jobs	61	male	91.1

```
write.table(datatoexport, file="ExportedFileName.txt", sep="\t")
```

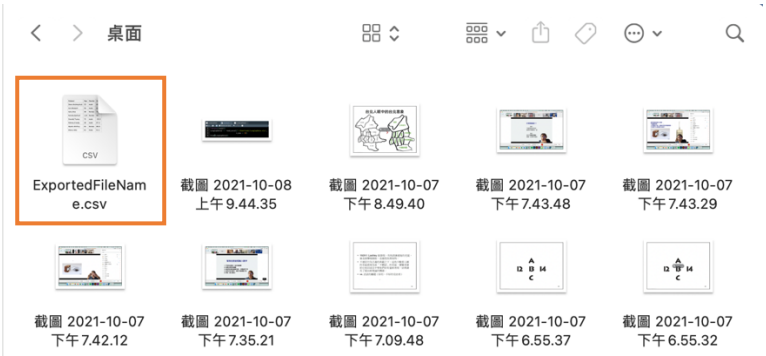



```
write.table(datatoexport, file="ExportedFileNameSpace", sep=" ")
```



*save in different working directory: file 要變成檔案路徑+要存成的檔名

```
write.table(datatoexport, file="/Users/chu/desktop/ExportedFileName.csv", row.names=F, sep=",")
```



(2) use “write.csv” command to save csv files:

*就不用 sep 了，但 write.table 還是比較方便因為可以存其他格式（.txt, space 等）

```
write.csv(datatoexport, file="/Users/chu/desktop/ExportedFileName.csv", row.names=F)
```

1.8 Working with Variables and Data in R

* To extract variables: (1) $ dollar sign (2) attach the dataset to R’s memory

- **Calculate the Mean**

```
> mean(Age)
錯誤發生在 mean(Age)：找不到物件 'Age'
```

```
> mean(LungCapData$Age)
[1] 12.3269
```

```
> attach(LungCapData)
> mean(Age)
[1] 12.3269
```

* attach 的相反：detach，從 R’s memory 裡去除

- **Ask R what type of variable it is**

```
> names(LungCapData)
[1] "LungCap"   "Age"       "Height"    "Smoke"     "Gender"    "Caesarean"
> class(LungCap)
[1] "numeric"
>
> class(Age)
[1] "numeric"
>
> class(Smoke)
[1] "character"
```

- **Ask R for a generic summary of the data**

```
> summary(LungCapData)
    LungCap            Age            Height        Smoke              Gender
 Min.   : 0.507   Min.   : 3.00   Min.   :45.30   Length:725         Length:725
 1st Qu.: 6.150   1st Qu.: 9.00   1st Qu.:59.90   Class :character   Class :character
 Median : 8.000   Median :13.00   Median :65.40   Mode  :character   Mode  :character
 Mean   : 7.863   Mean   :12.33   Mean   :64.84
 3rd Qu.: 9.800   3rd Qu.:15.00   3rd Qu.:70.30
 Max.   :14.675   Max.   :19.00   Max.   :81.80
  Caesarean
 Length:725
 Class :character
 Mode  :character
```

- **Convert a variable to a categorical variable or a factor**

```
> x = c(0,1,0,1,1,1,1,0)
> class(x)
[1] "numeric"
> summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.000   1.000   0.625   1.000   1.000
```

```
> x = as.factor(x)
> class(x)
[1] "factor"
> summary(x)
0 1
3 5
```

1.9 Subsetting Data in R using Square Brackets

- **Ask R the dimension of the data**

```
> dim(LungCapData)
[1] 725   6
```

*Row:725, Column:6

- **Ask the number of observations in a variable or a vector**

```
> length(Age)
[1] 725
```

* Variable "Age" consists of 725 observations

- **Applications of Square Brackets**

```
> Age[11:14]
[1] 19 17 12 10
```

* 取出某個 variable 的某個區間

```
> LungCapData[11:14, ]
# A tibble: 4 × 6
  LungCap   Age Height Smoke Gender Caesarean
    <dbl> <dbl>  <dbl> <chr> <chr>  <chr>
1   11.5     19   76.4 no    male   yes
2   10.9     17   71.7 no    male   no
3    6.52    12   57.5 no    male   no
4    6       10   61.1 no    female no
```

* 取出 Matrix 或 Dataframe 的一部份，像這裡是取 row 11–14，所有 column

- **Subsetting data based on values of other variables in the dataset**

Example 1. Calculate the mean age but only for females

```
> mean(Age[Gender=="female"])
[1] 12.44972
> mean(Age[Gender=="male"])
[1] 12.20708
```

Example 2. Create a subset of the data containing information for only one gender

```
> FemData <- LungCapData[Gender=="female", ]
> MaleData <- LungCapData[Gender=="male", ]
```

Example 3. Create a subset of the data for males who are over 15 years old

```
> attach(LungCapData)
> MaleOver15 <- LungCapData[Gender=="male" & Age>15, ]
> dim(MaleOver15)
[1] 89  6
> MaleOver15[1:4, ]
   LungCap Age Height Smoke Gender Caesarean
11  11.500  19   76.4    no   male       yes
12  10.925  17   71.7    no   male        no
23  10.025  16   72.4    no   male        no
40  11.325  17   77.7    no   male        no
```

1.10 Logic Statements, cbind and rbind Functions

* We will restrict ourselves for only looking the first five observations, so that we can see all the data on the screen and observe what's happening.

- **Logic statements**

```
> Age[1:5]
[1]  6 18 16 14  5
```

* 後續會用來操作的 data（1–5 列）

```
> temp <- Age>15
> temp[1:5]
[1] FALSE  TRUE  TRUE FALSE FALSE
```

* Age 是否大於 15

(2) use "as.numeric" command to have R return TRUE/FALSE as 1&0

```
> temp2 <- as.numeric(Age>15)
> temp2[1:5]
[1] 0 1 1 0 0
```

(3) use **multiple logical statements** within an R command to have a logical vector answering multiple questions

Example. create a vector indicating those who are female and smoke

```
> LungCapData[1:5, ]
  LungCap Age Height Smoke Gender Caesarean
1   6.475   6   62.1    no   male        no
2  10.125  18   74.7   yes female        no
3   9.550  16   69.7    no female       yes
4  11.125  14   71.0    no   male        no
5   4.800   5   56.9    no   male        no
```

* 後續會用來操作的 data（1–5 列）

```
> FemSmoke <- Gender=="female" & Smoke=="yes"
> FemSmoke[1:5]
[1] FALSE  TRUE FALSE FALSE FALSE
```

- **"cbind" & "rbind" Command**

  * We can attach vectors or matrices in a column-wise function using the "cbind" command, attach them in a row-wise fashion using the "rbind" command.

  Example. Attach "FemSmoke" variable to the entire dataset using the "cbind" command

```
> MoreData <- cbind(LungCapData, FemSmoke)
> MoreData[1:5, ]
  LungCap Age Height Smoke Gender Caesarean FemSmoke
1   6.475   6   62.1    no   male        no    FALSE
2  10.125  18   74.7   yes female        no     TRUE
3   9.550  16   69.7    no female       yes    FALSE
4  11.125  14   71.0    no   male        no    FALSE
5   4.800   5   56.9    no   male        no    FALSE
```

1.11 Setting Up Working Directory

* Working directory: one spot for saving all of our work. It's a good idea to create a different working directory for each project that we work on.

- **To know the current working directory**

```
> getwd()
[1] "/Users/chu/110-1/數量方法"
```

- **To set the working directory**

(1) use the "setwd" command and **specify the path to the folder** we would like to set as our working directory.

```
> setwd("/Users/chu/desktop/Project1")
>
> setwd("~/desktop/Project1")
```

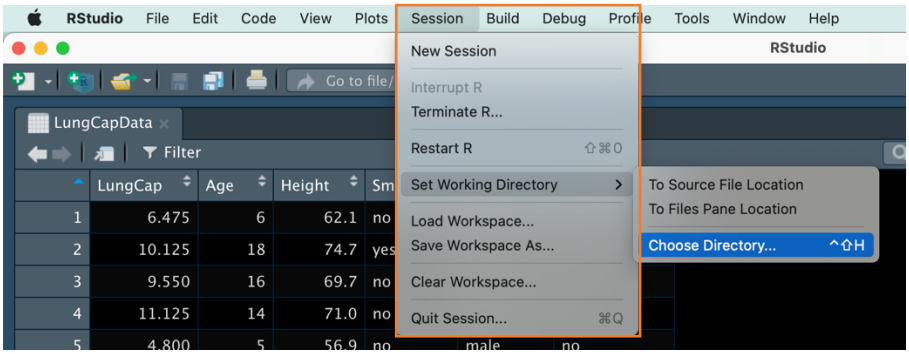
* 要給路徑！記得上下引號！
第二種是前面省略的寫法

(2) **create an object for saving the path to the folder**, and use the "setwd" command

```
> projectWD <- "/Users/chu/desktop/Project1"
> setwd(projectWD)
```

* 聰明做法

(3) Use menu: Session > Set working directory > Choose directory



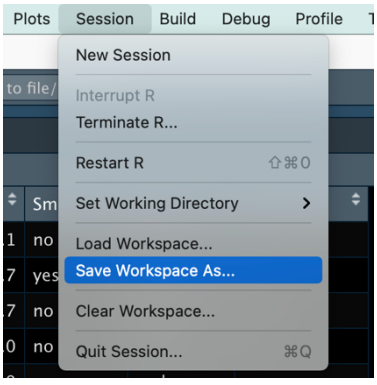
- **To save the workspace**

(1) use the "save.image" command

```
> save.image("FirstProject.Rdata")
```

* 記得設定副檔名為 Rdata，他才知道是要存 workspace

(2) Use menu: Session > Save Workspace As…

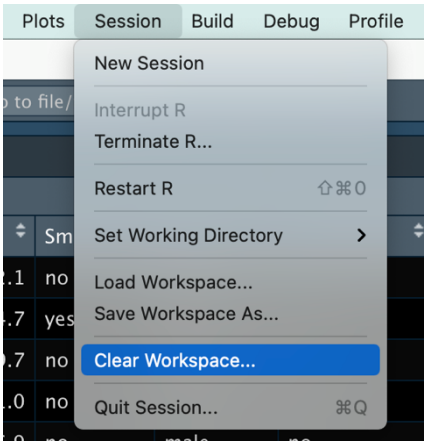


- **To clear the workspace and quit R**

  (1) remove all items from the current workspace

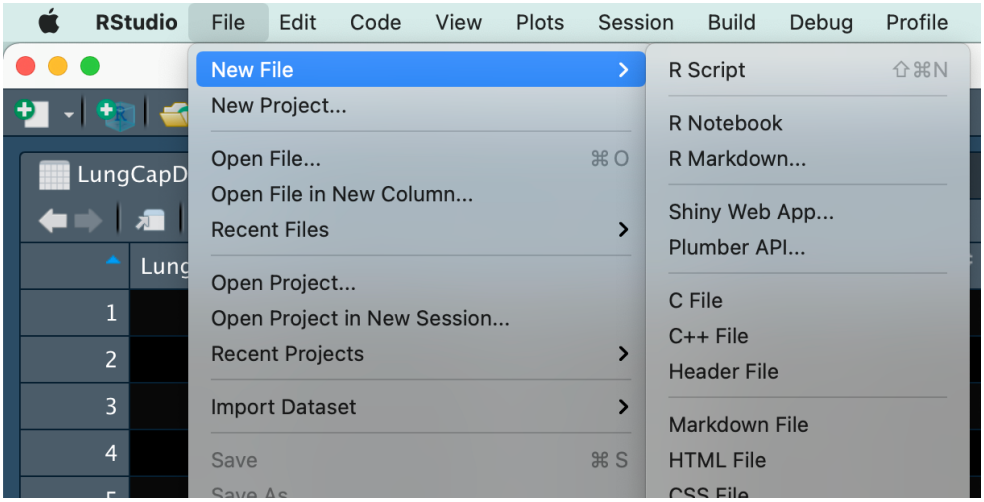
```
> rm(list=ls())
```

  (2) Use menu: Session > Clear Workspace



1.12 Writing Scripts in R

- Create a new script: Menu > New File > R Script



- 執行某幾行 code 的快捷鍵：command + enter
- Create a comment section: Menu > Code > Comment /Uncomment Lines
- Tab key helps find the name of commands or objects

1.14 Customizing the Look of Rstudio

(1) 選工具列的 tools > options (2) Rstudio menu > preferenes (Mac only)重要功能:

- 設定 working directory

- 設定是否自訂補齊（小括號中括號）
- 設定字型、字體大小、介面主題
- 設定介面 layout